**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ**

**Кафедра дифференциальных уравнений и системного анализа**

**Обнаружение сообществ алгоритмом Лейдена**

Курсовая работа

Яблонской Анны Олеговны

студентки 3 курса, специальность 1-31 03 09 Компьютерная математика   
и системный анализ

Научный руководитель:  
кандидат физ.-мат. наук,  
доцент А.Э. Малевич

Минск, 2022

**Оглавление**

[введение 3](#_Toc102393486)

[Глава 1 Начальные сведения 4](#_Toc102393487)

[1.1 Кластеризация и поиск сообществ 4](#_Toc102393488)

[1.2 Основные понятия и обозначения 4](#_Toc102393489)

[1.3 Модульность 6](#_Toc102393490)

[1.4 CMP 7](#_Toc102393491)

[1.5 Параметр разрешения 8](#_Toc102393492)

[1.6 Алгоритмы обнаружения сообществ 8](#_Toc102393493)

[Глава 2 алгоритм лувена 9](#_Toc102393494)

[Глава 3 Алгоритм лейдена 11](#_Toc102393495)

[Глава 4 Реализация алгоритма лейдена 14](#_Toc102393496)

[Глава 5 Примененте алгоритма лейдена 16](#_Toc102393497)

[заключение 17](#_Toc102393498)

[Список использованной литературы 18](#_Toc102393499)

введение

# Начальные сведения

## Кластеризация и поиск сообществ

Кластеризация — это метод машинного обучения, при котором объекты группируются таким образом, чтобы объекты из одного кластера были более похожи друг на друга, чем на объекты из других кластеров по какому-либо признаку.

Обнаружение сообществ — задача, которая старается найти лучшее разбиение графа на сообщества. Она немного отличается от кластеризации. Например кластеризация проводится на основе нескольких атрибутов, когда обнаружение сообществ основывается только на одном — ребра.

## Основные понятия и обозначения

граф с узлами и ребрами. Задавать графы можно различными способами: указанием множества вершин и ребер, графически, с помощью матрицы смежности или матрицы инцидентности. В данной работе графы будут задаваться при помощи матриц смежности A, для которой Ai,j=1, если между вершинами существует ребро, и 0, если ребра нет. Для взвешенных графов вес ребра определим как wi,j. Еслі граф невзвешенный, то wi,j=1.

— сообщество.

Разбиение графа: состоит из сообществ, где каждое сообщество Ci Vсостоит из множества узлов , для любых . Для двух множеств S и R будем использовать следующие обозначения:

—разбиение, которое получается при перемещении узла в сообщество разбиения .

Обозначим как подграф, образованный сообществом, т.е. и . Сообщество называется связным, если связный граф. И наоборот, сообщество называется несвязным, если несвязный граф.

Граф называется базовым. агрегированный граф базового графа . Узлы агрегированного графа являются сообществами в разбиении базового графа , то есть . Ребра агрегированного графа являются многогранниками. Количество ребер между двумя узлами агрегированного графа равняется числу ребер между узлами в двух соответствующих сообществах базового гафа . Следовательно,где мультимножество.

Мощность множества : , где если s не является множеством.

Функция качества или сообщества графа . Функция качества должна обладать свойством: , где обозначает одноэлементное разбиение агрегированного графа . Это гарантирует, что функция качества дает одинаковые результаты для базового и агрегированного графа.

Изменение функции качества после перемещения узла в сообщество обозначим, как . Другими словами: . Аналогично, обозначим изменение функции качества после перемещения множества узлов в сообщество: . Пустое сообщество обозначим . Следовательно, изменение функции качества после перемещения множества узлов в пустое, т.е. новое, сообщество: .

Операция уплощения(сглаживания) для множества S определяется следующим образом:

Где , если не является множеством. Множество, которое было сглажено называется уплощенным(сглаженным).

Также нас понадобится операция уплощения разбиения :

Т.е. уплощение разбиения – это операция в которой каждое сообщество сглажено. Разбиение, к которому была применена операция уплощения, называется уплощенным (сглаженным).

## Модульность

Одним из способов нахождения сообществ является модульность. Этот метод максимизирует разницу между фактическим и ожидаем количеством ребер в сообществе C. Связи внутри сообществ должны быть относительно частыми, в то время как связи между сообществами – редкими. Исходя из этого мы будем “поощрять” связи в сообществах и штрафовать за отсутствие связей в сообществе(1):

(1)

где , если и 0 в противном случае, веса . Минимальное значение данной функции соответствует наилучшему разделению. Выбор весов очень важен, т.к. они оказывают основное влияние на то, какие сообщества будут выбраны.

Существуют различные виды модульности, однако нам интересна лишь Константная модель Поттса(CMP), поскольку именно она применяется как функция качества в алгоритмах Лувенского и Лейдена.

## CMP

Для начала определим веса модели : , где это параметр разрешения, получаем (2):

(2)

Если мы определим количесво ребер в сообществе как (3),

(3)

a количество вершин /

Тогда уравнение (2) можно переписать в следующем виде(4):

(4)

Получаем что модель максимизирует число внутренних ребер, сохраняя небольшие сообщества. Параметр разрешения выполняет роль порога. Например, если сообщество состоит из ребер и вершин, тогда его лучше разделить на два сообщества R и S, если , где количество ребер между сообществами R и S. Такое соотношение определяет плотность связей между сообществами R и S. Получаем, что плотность связи между сообществами должна быть ниже , а внутри сообществ выше . Более высокой значение приводит к большему количеству сообществ и наоборот.

## Параметр разрешения

Разберем параметр подробнее, поскольку от выбора данного параметра зависит качество разбиения графа на сообщества.

В общем случае соответствует решению, когда все вершины находятся в одном сообществе. В тоже время соответствует решению только тогда, когда каждая вершина сама по себе является сообществом. Таким образом параметр разрешения необходимо выбирать из отрезка . Однако какой именно выбрать параметр?

Для разных моделей модульности следует выбирать различные параметры разрешения и оно зависит от количества кликов. Кликой называется подмножество вершин графа, любые две из которых соединены ребром. В общем случае для CMP стоит выбирать такой разрешающий параметр, что , где число кликов.

## Алгоритмы обнаружения сообществ

Существуют различные алгоритмы обнаружения сообществ. Наиболее популярными являются алгоритм Лувена и алгоритм Лейдена. Оба этих алгоритма основаны на модульности и рассматриваются как методы быстрого обнаружения сообществ в сетей. Более подробно данные алгоритмы будут рассмотрена в следующих главах.

# алгоритм лувена

Алгоритм Лувена был предложен в 2008 году как метод быстрого развертывания сообщества для больших сетей. Он основан на оптимизации функции модульности. Он очень популяр из-за простоты реализации и скорости. Рассмотрим сам алгоритм

1. Обычно алгоритм начинается с одноэлементного раздела, т.е. каждый узел является сообществом(однако можно запустить алгоритм и с другим разделение. Например, чтобы получить лучший результат, можно выполнить несколько последовательных итераций алгоритма и, используя разделения на сообщества из предыдущих итераций.)
2. Локальное перемещение узлов, в котором отдельные узлы перемещаются в сообщество, дающее наибольшее увеличение функции качества
3. Агрегация сети. На данном этапе каждое полученное сообщество, построенное на этапе 2 становится узлом и объединяет ребра между сообществами в утолщенные ребра, т.е. многогранники
4. Повторяет шаг 2 и 3 до тех пор, пока функция качества не может увеличиваться

Алгоритм Лувенского может находить “плохо” связанные сообщества, т.е. сообщества которые не связаны внутренне. В частности, он находит несвязные сообщества. Получаем, что одна часть сообщества, может достигнуть другой, только через внешний путь, т.е. вершину, принадлежащую другому сообществу. Такая вершина, являющаяся мостом между разъединенными частями сообщества, сама может быть перемещена в другое сообщество. Удаление такого узла приводит к появлению несвязного сообщества. И части данного сообщества необязательно будут образовывать новые сообщества, наоборот зачастую они так и остаются в одном сообществе. Это объясняется тем, что вершины внутри такого сообщества могут быть связаны сильнее между собой, чем с другими вершинами за пределами сообщества. В этом заключается недостаток алгоритма Лувена: он может находить сколько угодно слабо связанных сообществ.

Еще одним недостатком является то, что данный алгоритм дает лишь одну гарантию: раздел на сообщества, для которых гарантируется невозможность объединения в сообщество. Другими словами, он гарантирует, что сообщества хорошо разделены. Более сильные гарантии можно получить, повторяя алгоритм, т.е. используя предыдущие разбиения в качестве отправной точки следующей итерации. Таким образом качество разделения будет увеличиваться до тех пор, пока алгоритм не сможет вносить дальнейшие изменения. На этом этапе гарантируется, что каждый узел оптимально распределен. Такой способ итерации алгоритма гарантирует две вещи: 1) никакие сообщества не могут быть объединены; 2) никакие узлы не могут быть перемещены. Однако повторение алгоритма лишь усугубляет проблему слабо связанных сообществ, поскольку теперь алгоритм использует свое же предыдущее разбиение. Т.е. после первой итерации алгоритма было получено некоторое разбиение. Далее на первом шаге следующей итерации алгоритм снова переместит отдельные узлы. Некоторые из узлов могут быть мостами между узлами сообщества и перемещая их появляются слабо связанные сообщества. Более того алгоритм не обладает механизмом выявления таких сообществ. Таким образом повторение алгоритма с одной стороны улучшает разделение сообществ, а с другой ухудшает его. Поскольку решить проблему слабо связанных сообществ фактически невозможно, был придуман еще один алгоритм, алгоритм Лейдена, который решает данную проблему.

# Алгоритм лейдена

После того как было обнаружено, что алгоритм Лувена находит плохо связанные сообщества был предложен новый алгоритм: алгоритм Лейдена, который частично основан на алгоритме Лувена и является его усовершенствованной версией алгоритма Лувена и решает данную проблему. В отличие от предыдущего алгоритма, алгоритм Лейдена гарантирует, что сообщества хорошо связаны.

Данный алгоритм состоит из трех фаз:

1. Быстрое локальное перемещение узлов. Этот этап аналогичен фазе локального перемещения узлов в алгоритме Лувена. Однако данный алгоритм уже не посещает все узлы до тех пор, пока они не перестанут перемещаться, тем самым посещая и те узлы, которые нельзя перемещать. В процедуре быстрого локального перемещения посещаются только узлы, окрестности которых изменились.
2. Уточнение разделения. На данном этапе алгоритм находит разбиение , которое является уточнением разбиения , полученного на этапе 1 алгоритма. Здесь сообщества из могут быть разделены на под сообщества
3. Агрегация сети. На этом этапе происходит агрегация графа с учетом уточненного разбиения.

Более подробно данный алгоритм рассмотрен в следующей главе.

Итерация алгоритма Лейдена, при котором разбиение не меняется, называется стабильной итерацией. После стабильной итерации алгоритм гарантирует, что:

1. все узлы локально оптимально назначены, т.е. нет отдельных узлов которые можно было бы переместить
2. все сообщества являются γ-плотными по подразбиениям, т.е. сообщество можно разделить на две части так, что: эти две части хорошо связаны и ни одна из частей не может быть отделена от своего сообщества; каждая часть сама также γ-плотна по подразбиениям. γ-плотность подразбиения не означает, что отдельные узлы локально оптимально назначены. Это означает только то, что отдельные узлы хорошо связаны со своим сообществом.

В случае алгоритма Лувена после стабильной итерации все последующие итерации также будут стабильными, значит дальнейшие улучшения невозможны. В отличие от алгоритма Лейдена, в котором после стабильной итерации могут быть дальнейшие улучшения в более поздних итерациях. Фактически, когда мы продолжаем повторять алгоритм Лейдена, он сойдется к разбиению, для которого гарантируется, что:

1. все сообщества равномерно γ-плотны, т.е. нет подмножеств сообщества, которые можно отделить от него. Равномерная γ-плотность означает, что независимо от того, как сообщество разделено на две части, эти две части всегда будут хорошо связаны друг с другом. Кроме того, если все сообщества в разделении равномерно γ-плотны, качество раздела не слишком далеко от оптимального.
2. все сообщества оптимальны для подмножества.

Данный алгоритм гарантирует, что:

1. все сообщества γ-разделимы;
2. все сообщества γ-связаны.

В этих свойствах γ относится к параметру разрешения в функции качества CMP. Свойство γ-разделения также гарантируется алгоритмом Лувена, т.к. в нем говорится, что нет сообществ, которые можно объединить. Такое свойство является несколько более сильным вариантом обычной связности.

1. Самой надежной гарантией алгоритма Лейдена является оптимальность подмножеств. Сообщество является оптимальным по подмножеству, если все подмножества сообщества локально оптимально назначены, т.е. никакое подмножество не может быть перемещено в другое сообщество. Из данной гарантии вытекают γ-плотность и все остальные свойства.

Глава 4  
Реализация алгоритма лейдена

Первая стадия алгоритма — это быстрое локальное перемещение узлов. Алгоритм обходит все вершины в случайном порядке. Для каждого узла определяется, можно ли повысить функцию качества, переместив его из текущего сообщества в другое. Сообщество выбирается исходя из изменения значения функции качества . Перемещение, при котором изменение функции качество максимально определяет сообщество, к которому принадлежит вершина. Переместив вершину в сообщество, алгоритм запоминает всех ее соседей и добавляет их в список для посещения. Так продолжается до тех пор, пока все вершины не будут посещены.

Следующим этапом является уточнение разделения. Изначально на этом этапе все вершины разделены так, что сами по себе образуют сообщества. Т.е. они образуют новое разбиение . Затем алгоритм локально объединяет узлы в . На этом этапе рассматриваются сообщества разбиения . Для каждого рассматриваются все узлы, хорошо с ним связанные. Узлы хорошо связаны с подмножеством, если вес ребер, соединяющих вершину с подмножеством, больше, чем произведение γ, степени вершины и разности между порядком подмножества и степенью вершины. Далее алгоритм рассматривает все подмножества сообщества, хорошо связанные с ним, которые выбираются по такому же принципу, как и хорошо связанные вершины. Из этих подмножеств выбирается случайное, в которое перемещается вершина. Степень случайности выбора сообщества определяется параметром . Как можно заметить, в отличие от фазы локального перемещения узлов, здесь узлы не обязательно сливаются с подмножеством, которое дает наибольший прирост функции качества. Вместо этого узел может быть объединен с любым подмножеством, для которого функция качества увеличивается. Чем больше увеличение функции качества, тем больше вероятность того, что подмножество будет выбрано. Случайность выбора сообщества позволяет более широко исследовать пространство разбиения. Слияния узлов, приводящие к уменьшению функции качества, не учитываются. Это делает этап уточнения более эффективным.

Важно отметить, что слияния выполняются только внутри каждого сообщества разделения . Кроме того, узел объединяется с сообществом в только в том случае, если оба они достаточно хорошо связаны со своим сообществом в . После завершения фазы уточнения сообщества в часто будут разделены на несколько сообществ в ,но не всегда.

Последним этапом алгоритма является агрегация графа. Она основана на уточненном разделении . Узлы агрегированного графа являются сообществами в разбиении . Ребра агрегированного графа являются мультимножествами:

Создавая совокупную сеть на основе , алгоритм имеет больше возможностей для идентификации высококачественных связей.

Глава 5  
Применение алгоритма лейдена

заключение

Список использованной литературы